What is claimed is:

1. A method for accessing a second application programming interface in a second computing environment from a first process operating in a first computing environment that cannot natively access the second application programming interface, the method comprising the steps of:

detecting a first function call made by a first process in the first computing environment;

based on the first function call, generating an encapsulated function call for transfer from the first computing environment to the second computing environment, the encapsulated function call containing a mapping of first function call parameter values useable in the first computing environment to first meta parameter values; and

transferring the encapsulated function call containing the first meta parameter values from the first computing environment to the second computing environment.

2. The method of claim 1 further including the steps of:

receiving an encapsulated response from the second computing environment, the encapsulated response containing second meta parameter values produced in the second computing environment from performance of at least one second function call in the second application programming interface that corresponds to the first function call detected in the first computing environment;

parsing the encapsulated response to map the second meta parameter values back to the first function call parameters usable by the first process in the first computing environment.

3. The method of claim 1 wherein the step of detecting a first function call in the first computing environment comprises the steps of:

accessing a first application programming interface definition defining first function calls useable by first processes operating in the first computing environment, the first function calls corresponding to at least one second function call defined in a second

application programming interface useable by second processes in the second computing environment.

4. The method of claim 3 wherein the first function calls in the first application programming interface are defined in a Java-based format useable by Java-based first processes and wherein the second function calls in the second application programming interface are defined in a C-based format useable by C-based second processes.

5. The method of claim 3 further comprising the step of automatically deriving the first application programming interface definition from an analysis of the second application programming interface definition.

6. The method of claim 3 wherein the step of generating an encapsulated function call comprises the steps of:

extracting the first function call parameters from the first function call using the first application programming interface; and

for each of the first function call parameters:

selecting a respective first application programming interface definition data structure in the first application programming interface that can represent the first function parameter as a first meta parameter; and

copying the first function call parameter into the respective first application programming interface definition data structure to produce a first meta parameter that contains the value of the first function call parameter.

7. The method of claim 6 wherein the first meta parameters contain strings that represent values of the first function call parameters.

8. The method of claim 2 wherein the step of parsing the encapsulated response comprises the steps of:

parsing the encapsulated response to detect second meta parameters;

for each of the second meta parameters detected:

selecting a respective first application programming interface definition data structure in the first application programming interface that can represent the second meta parameter as a first function call parameter; and

copying second meta parameter values into the respective first application programming interface definition data structure to produce a first function call parameter that contains the value of the second meta parameter.

9. The method of claim 8 wherein at least one second meta parameter in the encapsulated response identifies a first function call definition within the first application programming interface definition, the first function call definition indicating the respective first application programming interface definition data structures that can accept the copied second meta parameter values.

10. The method of claim 8 wherein the second meta parameter values represent values of parameters processed by second function calls in the second application programming interface in the second computing environment and wherein the first application programming interface definition data structures that accept the copied second meta parameter values are natively incompatible for use in the second computing environment.

11. The method of claim 10 wherein the first application programming interface definition data structures in the first application programming interface are defined in a Java-based format useable by Java-based first processes and wherein the second function calls that operate to produce the second meta parameter values in the second application programming interface are defined in a C-based format useable by C-based second processes operating in the second programming environment.

12. A method for automatically generating applications allowing operation of an application programming interface in a second computing environment from a first process in a first computing environment, in which the first process is not natively compatible with the second computing environment, the method comprising the steps of:

analyzing an second application programming interface definition associated with a second computing environment to discover second function definitions in the second application programming interface definition;

based on the step of analyzing, automatically generating, for each second function definition discovered in the second application programming interface definition:

a first string generator capable of receiving a first function call in the first computing environment from a first process and encapsulating the first function call into an encapsulated function call;

a second parser capable of receiving the encapsulated function call and parsing the encapsulated function call to invoke a corresponding second function definition in the second application programming interface definition for operation within the second computing environment;

a second string generator capable of receiving an output from the second function call from a second process in the second computing environment and encapsulating the output into an encapsulated response; and

a first parser capable of receiving the encapsulated response and parsing the encapsulated response to return the output to the first function in the first process operating in the first computing environment.

13. The method of claim 12 wherein the steps of generating a first parser comprise the steps of:

receiving a first application programming interface specific grammar produced as a result of the step of analyzing the second application programming interface definition; and

processing the first application programming interface specific grammar using a first application programming interface processor to produce the first parser by

converting first function call definitions in the first application programming interface specific grammar into parser routines that can accept and parse meta parameters within encapsulated responses to provide first function call parameters back to first function calls associated with the first process that can operate in the first computing environment.

14. The method of claim 12 wherein the steps of generating a second parser comprise the steps of:

receiving a second application programming interface specific grammar produced as a result of the step of analyzing the second application programming interface definition; and

processing the second application programming interface specific grammar using a grammar processor to produce the second parser by converting second function call definitions in the second application programming interface specific grammar into parser routines that can accept and parse meta parameters within encapsulated function calls to provide second function call parameters to second function calls associated with the second process that can operate in the second computing environment.

15. The method of claim 12 wherein the step of generating a first string generator comprises the steps of:

for each second function call definition discovered in the second application programming interface definition:

producing a set of first application programming interface definition data structures that define first function call parameters that correspond to second function call parameters associated with the second function call definition; and

producing a set of first meta parameters that can represent the first application programming interface definition data structures; and

producing a first string generator function that can receive, from a first process that can operate in the first computing environment, a first function call that corresponds to the second function call definition, the

first string generator function capable of mapping the first function call parameters provided by the first function call to respective first meta parameters for encapsulation within an encapsulated function call for transfer to the second computing environment.

16. The method of claim 12 wherein the step of generating a second string generator comprises the steps of:

for each second function call definition discovered in the second application programming interface definition:

producing a set of second meta parameters that can represent second function call parameters used by the second function call definition by accessing second application programming interface definition data structures defined in the second application programming interface definition; and

producing a second string generator function that can receive, from a second process that can operate in the second computing environment, second function call parameters produced as output from the second process performing the second function call, the second string generator function capable of mapping the second function call parameters provided from invocation of the second function call to second meta parameters for encapsulation within an encapsulated response for transfer to the first computing environment.

17. The method of claim 12 further comprising the steps of:

analyzing the second application programming interface definition associated with the second computing environment to discover second data definitions in the second application programming interface definition; and

generating, for each second data definition discovered in the application programming interface definition, a first application programming interface definition

data structure capable of representing that second data definition in the second computing environment.

18. The method of claim 17 wherein:

the second computing environment is a C-based computing environment and wherein the second application programming interface definition is C-based and defines a interface to access functionality associated with a second process; and

the first computing environment Java-Based computing environment and wherein the second process is a Java-based process requiring access to the second function defined in the second application programming interface definition.

19. The method of claim 18 wherein the second application programming interface defines a set of second function definitions that provide access, via a second process, to data in a data storage system resource.

20. A computer system, comprising:

an interface;

a memory;

a processor; and

an interconnection mechanism coupling the interface, the processor and the memory;

wherein the memory is encoded with a first application programming interface string generator application and a first parser application that, when performed on the processor, produce a respective first application programming interface string generator process and first parser process that cause the computer system to access a second application programming interface in a second computing environment by a first process operating in the computer system by causing the computer system to perform the operations of:

detecting a first function call made by a first process in the first computing environment;

based on the first function call, generating an encapsulated function call for transfer via the interface from the first computing environment to the second computing environment, the encapsulated function call containing a mapping of first function call parameter values useable in the first computing environment to first meta parameter values; and

transferring the encapsulated function call containing the first meta parameter values from the first computing environment to the second computing environment via the interface.

21. The computer system of claim 20 wherein the first parser process further causes the computer system to perform the operations of:

receiving an encapsulated response from the second computing environment, the encapsulated response containing second meta parameter values produced in the second computing environment from performance of at least one second function call in the second application programming interface that corresponds to the first function call detected in the first computing environment;

parsing the encapsulated response to map the second meta parameter values back to the first function call parameters usable by the first process in the first computing environment.

22. The computer system of claim 20 wherein when the computer system performs the operation of detecting a first function call in the first computing environment, the computer system further performs the operation of:

accessing a first application programming interface definition defining first function calls useable by first processes operating in the first computing environment, the first function calls corresponding to at least one second function call defined in a second application programming interface useable by second processes in the second computing environment.

23. The computer system of claim 22 wherein the first function calls in the first application programming interface are defined in a Java-based format useable by Java-based first processes and wherein the second function calls in the second application programming interface are defined in a C-based format useable by C-based second processes.

24. The computer system of claim 22 wherein the memory is encoded with a grammar application programming interface processor that when operated on the processor further causes the computer system to perform an operation of automatically deriving the first application programming interface definition from an analysis of the second application programming interface definition.

25. The computer system of claim 22 wherein when the computer system performs the operation of generating an encapsulated function call, the computer system performs the operations of:

extracting the first function call parameters from the first function call using the first application programming interface; and

for each of the first function call parameters:

selecting a respective first application programming interface definition data structure in the first application programming interface that can represent the first function parameter as a first meta parameter; and

copying the first function call parameter into the respective first application programming interface definition data structure to produce a first meta parameter that contains the value of the first function call parameter.

26. The computer system of claim 25 wherein the first meta parameters contain strings that represent values of the first function call parameters.

27. The computer system of claim 21 wherein when the computer system performs the operation of parsing the encapsulated response, the computer system performs the operations of:

parsing the encapsulated response to detect second meta parameters;

for each of the second meta parameters detected:

selecting a respective first application programming interface definition data structure in the first application programming interface that can represent the second meta parameter as a first function call parameter; and

copying second meta parameter values into the respective first application programming interface definition data structure to produce a first function call parameter that contains the value of the second meta parameter.

28. The computer system of claim 27 wherein at least one second meta parameter in the encapsulated response identifies a first function call definition within the first application programming interface definition, the first function call definition indicating the respective first application programming interface definition data structures that can accept the copied second meta parameter values.

29. The computer system of claim 27 wherein the second meta parameter values represent values of parameters processed by second function calls in the second application programming interface in the second computing environment and wherein the first application programming interface definition data structures that accept the copied second meta parameter values are natively incompatible for use in the second computing environment.

30. The computer system of claim 29 wherein the first application programming interface definition data structures in the first application programming interface are defined in a Java-based format useable by Java-based first processes and wherein the

second function calls that operate to produce the second meta parameter values in the second application programming interface are defined in a C-based format useable by C-based second processes operating in the second programming environment.

31. A computer system comprising:

a memory;

a processor; and

an interconnection mechanism coupling the processor and the memory;

wherein the memory is encoded with a grammar application programming interface processor application that, when performed on the processor, produces a grammar application programming interface processor that cause the computer system to automatically generate applications allowing operation of an application programming interface in a second computing environment from first process in a first computing environment, in which the first process is not natively compatible with the second computing environment, by causing the computer system to perform the operations of:

analyzing an second application programming interface definition associated with a second computing environment in the memory to discover second function definitions in the second application programming interface definition;

based on the step of analyzing, automatically generating, for each second function definition discovered in the second application programming interface definition:

a first string generator capable of receiving a first function call in the first computing environment from a first process and encapsulating the first function call into an encapsulated function call;

a second parser capable of receiving the encapsulated function call and parsing the encapsulated function call to invoke a corresponding second function definition in the second application programming interface definition for operation within the second computing environment;

a second string generator capable of receiving an output from the second function call from a second process in the second computing environment and encapsulating the output into an encapsulated response; and

a first parser capable of receiving the encapsulated response and parsing the encapsulated response to return the output to the first function in the first process operating in the first computing environment.

32. The computer system of claim 31 wherein when the computer system performs the operation of generating a first parser, the computer performs the operations of:

receiving a first application programming interface specific grammar produced as a result of the step of analyzing the second application programming interface definition; and

processing the first application programming interface specific grammar using a first application programming interface processor to produce the first parser by converting first function call definitions in the first application programming interface specific grammar into parser routines that can accept and parse meta parameters within encapsulated responses to provide first function call parameters back to first function calls associated with the first process that can operate in the first computing environment.

33. The computer system of claim 31 wherein when the computer system performs the operation of generating a second parser, the computer system performs the operations of:

receiving a second application programming interface specific grammar produced as a result of the step of analyzing the second application programming interface definition; and

processing the second application programming interface specific grammar using a grammar processor to produce the second parser by converting second function call definitions in the second application programming interface specific grammar into parser routines that can accept and parse meta parameters within encapsulated function calls to provide second function call parameters to second function calls associated with the second process that can operate in the second computing environment.

34. The computer system of claim 31 wherein when the computer system performs the operation of generating a first string generator, the computer system performs the operations of :

for each second function call definition discovered in the second application programming interface definition:

producing a set of first application programming interface definition data structures that define first function call parameters that correspond to second function call parameters associated with the second function call definition; and

producing a set of first meta parameters that can represent the first application programming interface definition data structures; and

producing a first string generator function that can receive, from a first process that can operate in the first computing environment, a first function call that corresponds to the second function call definition, the first string generator function capable of mapping the first function call parameters provided by the first function call to respective first meta parameters for encapsulation within an encapsulated function call for transfer to the second computing environment.

35. The computer system of claim 31 wherein when the computer system performs the operation of generating a second string generator, the computer system performs the operations of:

for each second function call definition discovered in the second application programming interface definition:

producing a set of second meta parameters that can represent second function call parameters used by the second function call definition by accessing second application programming interface definition data structures defined in the second application programming interface definition; and

producing a second string generator function that can receive, from a second process that can operate in the second computing environment, second function call parameters produced as output from the second process performing the second function call, the second string generator function capable of mapping the second function call parameters provided from invocation of the second function call to second meta parameters for encapsulation within an encapsulated response for transfer to the first computing environment.

36. The computer system of claim 31 wherein the computer system further performs the operations of:

analyzing the second application programming interface definition associated with the second computing environment to discover second data definitions in the second application programming interface definition; and

generating, for each second data definition discovered in the application programming interface definition, a first application programming interface definition data structure capable of representing that second data definition in the second computing environment.

37. The computer system of claim 36 wherein:

the second computing environment is a C-based computing environment and wherein the second application programming interface definition is C-based and defines a interface to access functionality associated with a second process; and

the first computing environment Java-Based computing environment and wherein the second process is a Java-based process requiring access to the second function defined in the second application programming interface definition.

38. The computer system of claim 37 wherein the second application programming interface defines a set of second function definitions that provide access, via a second process, to data in a data storage system resource.

39. A computer system comprising:

a memory;

a processor; and

an interconnection mechanism coupling the processor and the memory;

wherein the memory is encoded with a grammar application programming interface processor application that, when performed on the processor, produces a means for automatically generating applications allowing operation of an application programming interface in a second computing environment from first process in a first computing environment, in which the first process is not natively compatible with the second computing environment, by providing the computer system with:

means for analyzing an second application programming interface definition associated with a second computing environment in the memory to discover second function definitions in the second application programming interface definition;

means, based on the means for analyzing, for automatically generating, for each second function definition discovered in the second application programming interface definition:

a first string generator capable of receiving a first function call in the first computing environment from a first process and encapsulating the first function call into an encapsulated function call;

a second parser capable of receiving the encapsulated function call and parsing the encapsulated function call to invoke a corresponding second function definition in the second application programming interface definition for operation within the second computing environment;

a second string generator capable of receiving an output from the second function call from a second process in the second computing environment and encapsulating the output into an encapsulated response; and

a first parser capable of receiving the encapsulated response and parsing the encapsulated response to return the output to the first function in the first process operating in the first computing environment.

40. A computer system, comprising:

an interface;

a memory;

a processor; and

an interconnection mechanism coupling the interface, the processor and the memory;

wherein the memory is encoded with logic instructions that, when executed on the processor in the computer system, provide the computer system with:

means for detecting a first function call made by a first process in the first computing environment;

means, based on the first function call, for generating an encapsulated function call for transfer via the interface from the first computing environment to the second computing environment, the encapsulated function call containing a mapping of first function call parameter values useable in the first computing environment to first meta parameter values; and

means for transferring the encapsulated function call containing the first meta parameter values from the first computing environment to the second computing environment via the interface.

41. A computer program product having a computer-readable medium including computer program logic encoded thereon, that when executed on a computer system having a coupling of a memory and a processor, provides a method for accessing a second application programming interface in a second computing environment from a first process operating in a first computing environment that cannot natively access the second application programming interface by causing the processor to perform the operations of:

detecting a first function call made by a first process in the first computing environment;

based on the first function call, generating an encapsulated function call for transfer from the first computing environment to the second computing environment, the encapsulated function call containing a mapping of first function call parameter values useable in the first computing environment to first meta parameter values; and

transferring the encapsulated function call containing the first meta parameter values from the first computing environment to the second computing environment.

42. A computer program product having a computer-readable medium including computer program logic encoded thereon, that when executed on a computer system having a coupling of a memory and a processor, provides a technique for automatically generating applications allowing operation of an application programming interface in a second computing environment from a first process in a first computing environment, in which the first process is not natively compatible with the second computing environment, by causing the processor to perform the operations of:

analyzing an second application programming interface definition associated with a second computing environment to discover second function definitions in the second application programming interface definition;

based on the step of analyzing, automatically generating, for each second function definition discovered in the second application programming interface definition:

a first string generator capable of receiving a first function call in the first computing environment from a first process and encapsulating the first function call into an encapsulated function call;

a second parser capable of receiving the encapsulated function call and parsing the encapsulated function call to invoke a corresponding second function definition in the second application programming interface definition for operation within the second computing environment;

a second string generator capable of receiving an output from the second function call from a second process in the second computing environment and encapsulating the output into an encapsulated response; and

a first parser capable of receiving the encapsulated response and parsing the encapsulated response to return the output to the first function in the first process operating in the first computing environment.

43. A method for accessing a second application programming interface in a second computing environment from a first computing environment that operates a first process that cannot natively access the second application programming interface, the method comprising the steps of:

receiving an encapsulated function call containing first meta parameters from a first computing environment;

parsing the encapsulated function call to map the first meta parameters into second function call parameters associated with a second function definition in the second application programming interface definition;

invoking operation of the second function associated with the second process;

receiving second function call parameter values as output from invocation of second function;

mapping the second function call parameters into an encapsulated response; and

transferring the encapsulated response to the first computing environment.

44. The method of claim 43 wherein:

the first process operating in the first computing environment is a Java-based families this of them about process;

the second function associated with the second process is a C- based function; and

the first meta parameters represent an interpretation of Java-based data structures provided in response to the first process invoking a first function call.

45. The method of claim 43 wherein the first meta parameters contain strings that represent values of the first function call parameters.

46. The method of claim 43 wherein the encapsulated response includes at least one second meta parameter that contains a mapping of second function call parameters output from invocation of the second function.